Reinforcement learning algorithms in highway driving

Thesis from

Antonios Iosifidis
(AM: 1335)

Supervisor: Associate Prof. Nikolaos Ploskas

Department of Electrical & Computer Engineering,
University of Western Macedonia
Date: 20.01.2024

Contents

List of Figures

Copyright

I explicitly declare that according to Article 8 of the law 1599/1986 and Articles 2, 4, and 6 par. Three of the law 1256/1982, this thesis entitled End-to-End approach for autonomous driving on the high way, as well as the files and source codes developed or modified in the context of this thesis and explicitly mentioned in the associated text and which has been conducted, implemented and produced at the Department of Electrical & Computer Engineering of the University of Western Macedonia, under the supervision of the Associate Prof. Nikolaos Ploskas is exclusively a product of personal work. It does not violate any form of copyright of third parties and is not a product of partial or total copying and the sources used are limited to bibliographical references only. The parts where I have used ideas, text, files and/or sources of other authors are clearly and completely indicated in the text with appropriate citations and the relevant references are included in the bibliographical references section with complete and detailed description. Copy, storage and distribution of this study, completely or partially, for commercial purposes is prohibited. Reproduction, storage and distribution for non-profit, educational or research purposes is permitted, provided the source is acknowledged and this message is retained. Questions regarding use of the study for profit should be directed to the author. The viewpoints and inferences contained in this study are exclusively those of the author.

Copyright (C) Antonios Iosifidis & Nikolaos Ploskas, 2024, Kozani, Greece.

Student Signature

Abstract

This thesis examines the potential of employing photorealistic synthetic data for training neural networks to achieve collision-free autonomous driving in highway environments. Centered around a case study of a highway accident involving a Tesla Model 3, the research utilizes Unreal Engine 5 to reconstruct the event and explore the effectiveness of an End-to-End autopilot system. The study innovatively applies Domain Structured Randomization to generate varied driving scenarios, assessing the autopilot's adaptability and response. The absence of real-world domain access underscores the significance of synthetic data in simulating and analyzing the incident, aiming to enhance the safety features of autonomous driving technologies.

Keywords: photorealistic synthetic data, neural networks, autonomous driving, highway environments.

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει τις δυνατότητες χρήσης φωτορεαλιστικών συνθετικών δεδομένων για την εκπαίδευση νευρωνικών δικτύων,ώστε να επιτευχθεί αυτόνομη οδήγηση χωρίς συγκρούσεις σε περιβάλλοντα αυτοκινητοδρόμων. Με επίκεντρο τη μελέτη περίπτωσης ενός ατυχήματος σε αυτοκινητόδρομο,στο οποίο ενεπλάκη ένα Tesla Model 3, η έρευνα χρησιμοποιεί την Unreal Engine 5 για την ανακατασκευή του συμβάντος και τη διερεύνηση της αποτελεσματικότητας ενός συστήματος αυτόματου πιλότου End-to-End. Η διπλωματική εργασία εφαρμόζει καινοτόμες μεθόδους Domain Structured Randomization για τη δημιουργία ποικίλων σεναρίων οδήγησης, αξιολογώντας την προσαρμοστικότητα και την απόκριση του αυτόματου πιλότου.Η έλλειψη πρόσβασης στον πραγματικό κόσμο υπογραμμίζει τη σημασία των συνθετικών δεδομένων για την προσομοίωση και την ανάλυση του συμβάντος, με στόχο την ενίσχυση των χαρακτηριστικών ασφαλείας των τεχνολογιών αυτόνομης οδήγησης.

Λέξεις κλειδιά: φωτορεαλιστικά συνθετικά δεδομένα, νευρωνικά δίκτυα, αυτόνομη οδήγηση, περιβάλλοντα αυτοκινητοδρόμων.

Acknowledgments

I am immensely grateful to my mother and father, whose continuous support throughout my academic journey has shaped my resilient character. Their constant encouragement to delve into diverse scientific domains and their belief in my capabilities have been the cornerstone of my perseverance. I extend my heartfelt appreciation to IAV and especially my supervisor, Robin Könhauser, for their faith in the idea of my thesis, for fostering an environment where my imagination thrived, and for their consistent support, never dampening my spirit. I extend my sincere gratitude to my university supervisor, Nikolaos Ploskas, whose invaluable guidance and scholarly feedback have been pivotal in shaping the academic and scientific approach of this thesis. Finally, I am grateful to my brother, whose steadfast support during challenging times has inspired me during my academic years. Also, emulating his work ethic has significantly influenced my approach to work, making him an exemplary role model in my professional life and in how I navigate life's challenges.

Chapter 1

Introduction

1.1 Motivation and background

Since the inception of the modern automotive era, vehicle safety has undergone a significant evolution. Innovations, such as safety belts, airbag sensing systems, and crash testing, have been instrumental. The information gained in these crash tests [6] has always played and continues to play a critical role in the design of new vehicles. Despite these enhancements, global road accidents still claim approximately 1.3 million lives annually, with human error being the leading cause [8]. In response, the intersection of rapidly advancing technology and the integration of machine learning algorithms in the automotive sector has paved the way for the development of autonomous driving. However, the journey towards entirely safe autonomous vehicles has its challenges. Incidents involving autonomous vehicles persist. Since replicating every potential real-world situation, such as accidents, is unfeasible, contemporary vehicle simulations can be used as a bridge to reality. Modern vehicle simulations are able to represent complete road environments with a realistic, graphical, and physical representation. Therefore, the primary concern of this thesis revolves around the effective implementation of a network, based on the End-to-End approach, that has been trained on synthetic data to function optimally within a recreation of a highway environment.

1.2 Problem statement

In this thesis, we focus on a highway self-driving car accident on Taiwan Highway 1 in 2020 with a Tesla Model 3 [7]. The sequence of events unfolds as follows: the collision occurred in the morning hours on National Highway 1, specifically at the 268.4-kilometer mark in the southbound lanes. The County Fire Bureau was alerted to the traffic incident involving the truck and a sedan at approximately 6:40 am. Findings by the Highway Police Bureau revealed that the driver of the Tesla, traveling at an estimated speed of 110 kilometers per hour, was allegedly on complete autopilot and couldn't prevent the collision with the roof of the overturned truck. Figure 1.1 depicts the the described crash phenomenon showcases. The primary visual evidence available consists of footage from security cameras. Therefore, without access to the real-world target domain [18], by using Unreal Engine 5, this thesis aims to provide an immersive and detailed driver's perspective of the events leading up to the crash.

Figure 1.1: Tesla Model 3 in the highway accident https://www.whichcar.com.au/car-news/tesla-model-3-drives-straight-into-a-truck

1.3 Aim and objectives

This thesis investigates if the creation of photo-realistic synthetic data in the context of highway driving can be used effectively. The research aims to assess the practicality of using neural networks, which have been trained and tested on synthetic data, to facilitate collision-free driving. A significant aspect of this study is the implementation of a TensorFlow NVIDIA autopilot that follows the End-to-End approach in the context of collision-free autonomous driving [1]. Additionally, the thesis extends the application of Structured Domain Randomization techniques [11] to create specific alternative driving scenarios. This is done to analyze the response of the autopilot system in varied and randomized environments. More precisely, the study generates a spectrum of alternative scenarios to evaluate the responsiveness and effectiveness of the autopilot in these diverse conditions. Through this approach, the thesis aims to contribute significantly to developing and enhancing autonomous driving technologies, particularly in ensuring safer driving experiences.

1.4 Research overview

The thesis structure is as follows. Chapter 2 reviews the technical and theoretical foundations for understanding the relevant content from the areas of autonomous driving cars, the end-to-end approach, and the help of synthetic data. In Chapter 3, the guiding question of this thesis is clearly addressed once again, and the experiment design takes place. In addition, the complete process and execution of synthetic data generation training of the neural network are described. Chapter 4 analyzes the definition of evaluation metrics and presents the final results. Finally, Chapter 5 summarizes the scientific work, gives an outlook on the topic, and suggests future updates.

Chapter 2

Theoretical and Technical Analysis

2.1 Automation levels for self-driving cars

The driver's primary responsibility is to maintain control over the vehicle, thereby facilitating its operation. Driver assistance technologies, including Adaptive Cruise Control (ACC), lane departure warning, and advanced parking capabilities, assume partial responsibility for both the lateral and longitudinal guiding of the vehicle, effectively reducing the need for complete driver control. This demonstrates that the responsibility for vehicle guiding and the vehicle's direction is a crucial component of automation, as outlined by the SAE J3016 standard [9]. In the initial stage of automation levels, it is presumed that there is a single driver who possesses complete control and accountability for guiding the vehicle. Throughout the various stages, the transfer of control and responsibility to the driver assistance systems is a continual process. Initially, the driver is given a supervisory role (stages 2-3), in which intervention is required if necessary. In the final stages (4-5), an active driver is no longer needed, and all vehicle occupants become passengers. Automated driving is achieved from level 4. A driver is no longer required in the active vehicle control loop from this level.

2.2 Autonomous driving approaches

2.2.1 Modular approach

The most classical approach that is used in the context of autonomous driving is the modular pipeline approach [14]. The modular approach divides the world system into three independent modules. The perception model takes the raw sensor information as input and tries to perceive the surrounding environment. The second module is the Planning module, which takes as input the representation of the scene and tries to compute the best trajectories for the vehicle while trying to avoid collisions. Finally, the last module is the Control module, which commands the steering wheel, the brake, and the accelerator pedal to follow the computed trajectory. Figure 2.1 depicts an overview of the described Modular Pipeline approach. In more detail, the perception [12] of the environment in the context of autonomous driving assistance systems refers to the electrical detection of the vehicle environment with the aid of corresponding sensor technology. The sensor data creates an image of the vehicle environment for route planning and decision-making. The data of several sensor types are combined for the most accurate representation of the vehicle environment. This is referred to as sensor data fusion. Four sensors for environment perception have become established in the automotive industry. Each sensor has advantages and disadvantages and, therefore, a specific application area. These sensors are the camera, the radar, the lidar, and the sonar. At the same time, path planning algorithms utilize optimization techniques to determine the most efficient trajectory, considering dynamic barriers, traffic conditions, and vehicle dynamics. The ability to predict the actions and movements of various road participants, such as automobiles, bicycles, and pedestrians, is essential in ensuring safety and effectiveness of autonomous driving systems. Using predictive modeling approaches, frequently based on machine learning examples, becomes necessary in this context. Recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks have been recognized as powerful tools for modeling temporal dependencies and predicting future trajectories of dynamic objects. By integrating contextual data obtained from sensors, the autonomous system can thoroughly comprehend the intentions and trajectories of nearby entities. The integration of independent actions with established regulations and expectations is a fundamental element of the decision-making process. To ensure safe and predictable interactions with cars controlled by humans, it is vital for autonomous vehicles to demonstrate compliance with regulatory norms.This involves thoroughly incorporating traffic legal rules into the decision-making algorithms and the capacity to comprehend and react to traffic signs, signals, and right-of-way protocols.

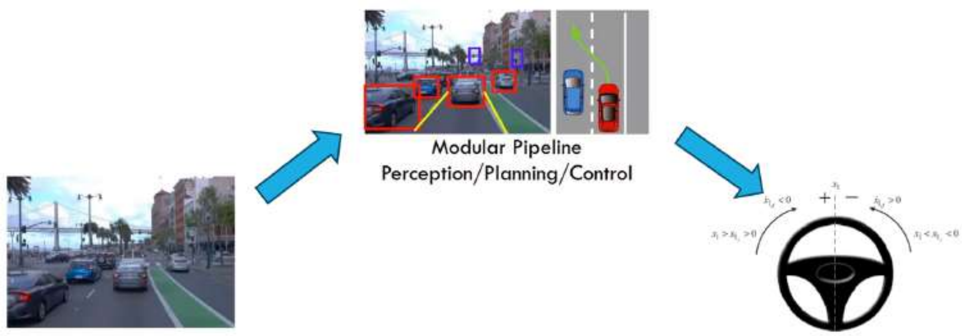

Figure 2.1: Modular Pipeline approach
https://uni-tuebingen.de/fakultaeten/mathematisch-naturwissenschaftliche-fakultaet/fachbereiche/informatik/lehrstuehle/autonomous-vision/lectures/self-driving-cars/

2.2.2 End-to-End approach

In the modular autonomous driving pipeline, as described before, some individual models function as distinct components, each dedicated to a specialized task. This modular architecture affords the benefit of simplified troubleshooting. Nonetheless, the optimization objectives are disparate across these modules. For instance, the perception module may focus on maximizing mean average precision (mAP), whereas the planning module prioritizes the safety and comfort of the driving experience. Consequently, this disjointed framework may not converge towards a cohesive goal. Specifically, the overarching objective of effective planning and control. Moreover, as the process unfolds in sequence, errors may cascade from one module to the next, culminating in a degradation of information for the driving system. Additionally, deploying multiple models to address various tasks increases the computational load, which could result in a sub-optimal allocation of processing resources. In contrast, an end-to-end (E2E) autonomous driving system presents several advantages over the modular approach. The most important benefit is the system's streamlined nature, which integrates perception, prediction, and planning within a singular model architecture that is easy to concurrent training. This holistic system is optimized to fulfill the ultimate driving task, with intermediate representations aligned with this end goal. Shared structural elements within the model enhance computational efficiency. Furthermore, the system's reliance on data-driven optimization promises to uncover innovative solutions that may surpass the capabilities of traditionally engineered systems. The category of End to End approach that was utilized in this research is Imitation learning. This approach is predicated on the concept of behavioral acquisition from expert demonstrators, whereby an autonomous vehicle's control policy is shaped by mimicking human driving behavior. The central idea of the imitation learning approach is the use of advanced machine learning techniques, typically neural networks, to interpret and learn from high-dimensional input data. These networks are trained on datasets of real-world driving scenarios, capturing the dynamic nature of the driving environment. The appeal of imitation learning within the End-to-end context lies in its potential to simplify the traditionally multi-layered architecture of autonomous systems. By directly correlating input data with expert responses, the system aims to develop a comprehensive driving policy that is both adaptable and resilient to the unpredictable variables present in real-world driving conditions. Figure 2.2 is a general showcase of the described End-to-End approach. The mathematical notations and types that Imitation learning follows are described further:

- The state $s \in S$ - The state s is an element of the state space S.
- The action $a \in A$ - The action a is an element of the action space A.
- The policy $\pi_\theta : S \to A$ - The policy π is a function mapping from the state space S to the action space A, parameterized by θ.

- The optimal action $a^* \in A$ - The optimal action a^* is an element of the action space A.
- The optimal policy $\pi^* : S \to A$ - The optimal policy π^* is a function mapping from the state space S to the action space A.
- The state dynamics $P(s_{i+1}|s_i, a_i)$ - The state dynamics P is a probability distribution over the next state s_{i+1}, given the current state s_i and action a_i.
- Often deterministic: $s_{i+1} = T(s_i, a_i)$ - The deterministic mapping of state dynamics where the next state s_{i+1} is determined by a transition function T, given the current state s_i and action a_i.
- Rollout: Given s_0, $\tau = (s_0, a_0, s_1, a_1, \ldots)$ - A rollout is a sequence or trajectory τ consisting of states and actions starting from the initial state s_0.
- Loss function: $\mathcal{L}(a^*, a)$ - The loss function $\mathcal{L}$ quantifies the loss of an action a given the optimal action a^*.

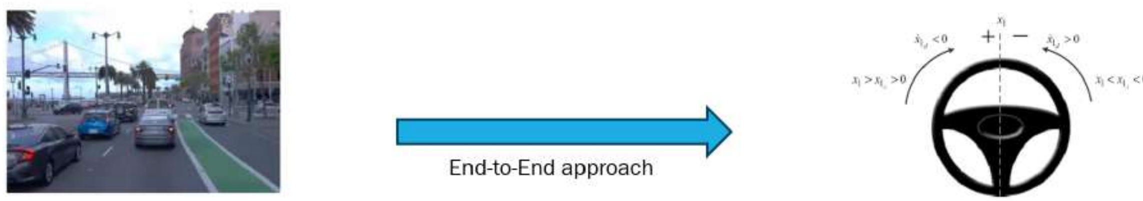

Figure 2.2: End-to-End approach
https://uni-tuebingen.de/fakultaeten/mathematisch-naturwissenschaftliche-fakultaet/fachbereiche/informatik/lehrstuehle/autonomous-vision/lectures/self-driving-cars/

2.3 Synthetic data

Synthetic data that mirrors real-world scenarios offers a lot of advantages. Firstly, it enables the simulation of diverse and complex driving environments impractical to replicate in reality. This feature is crucial for training and validating autonomous driving algorithms, where exposure to a wide range of scenarios is essential for robust performance. Moreover, synthetic data allows the exploration of alternative edge cases and the assessment of system responses under various conditions, thereby ensuring a comprehensive evaluation of autonomous driving systems. An important aspect is the optimization and validation of driver assistance systems during system integration into a vehicle. Real vehicle tests are still essential to the optimization and validation process of Engine control unit functions. They cover the entire vehicle and the Engine control unit network and include the driver and his driving behavior in these tests. However, the complexity of testing Vehicle Assistance Systems (VAS) functions on actual vehicles is increasing. These tests must account for the vehicle and a dynamic and challenging environment. Constructing such an environment in reality is a challenging and coordination-intensive task. It necessitates a setting that mirrors real-world traffic conditions, complete with its real-world unpredictability. This process often leads to substantial increases in personnel, materials, and costs. A simulated environment is used instead to reduce the effort required to create this real environment. In this approach, a real vehicle that drives in a simulated world is called a vehicle-in-the-loop (VIL). Various companies are considering implementing a simulation to create a development or validation environment [4].

2.4 Reality gap

Synthetic data generation has many advantages; however, a set of challenges need to be addressed for it to be effective. Synthetic data sets are generated using simulation; hence, we must close the gap between the simulation and the real world. This gap is called the domain gap, which can be divided into two parts: The appearance gap is the set of pixel-level differences between real and synthetic images. These differences can result from differences in object detail, materials, or, in the case of synthetic data, differences in the capabilities of the rendering system used. The content gap refers to the difference between the domains. This includes factors like the number of objects in the scene, the diversity in type and placement, and similar contextual information. The issue of transferring models from simulated environments to the physical world is often referred to as the 'reality gap' in the context of autonomous driving. The reality gap is primarily attributed to disparities in physical parameters and the inaccuracies in physical modeling, like the interaction between surfaces and assets.System identification is a famous approach in approaching the reality gap context. It involves constructing a mathematical representation of a physical system, with the simulator embodying this representation. However, accurately calibrating these simulators can be cost-prohibitive and complicated by varying physical parameters influenced by external conditions like temperature. Another approach is Domain Adaptation(DA) [3] [10], which entails updating the data distribution in a simulator to better align with the real world. This strategy, often implemented through adversarial loss or Generative Adversarial Networks (GAN), aims to bridge the gap between simulated and real environments. Domain Randomization (DR) [16] [15] is a particularly noteworthy technique in this regard. It involves generating many simulated scenarios with varying properties and training models to function effectively across this diverse range of conditions. This method increases the likelihood of the model adapting to the real-world environment, as it is anticipated to fall within the spectrum of the simulated scenarios. In summary, bridging the reality gap in autonomous driving through methods like System Identification, Domain Adaptation, and Domain Randomization is pivotal. These approaches enhance the realism of simulations, thereby facilitating the transfer of robust, adaptable models to the physical world, a critical step in advancing autonomous driving technologies.

Chapter 3

Methodology

3.1 Overview

The fundamental question of this scientific study will now be investigated by examining an experiment. The first main objective is to realistically recreate a digital twin of the car accident and produce alternative scenarios based on theoretical scientific assumptions about the cause of the accident. Then, the next objective is to implement and use an End-to-End approach for the navigation of the ego-car to see how it reacts in the original and alternative scenarios. The Nvidia PilotNet will be utilized for this objective. The version of PilotNet used in this work is a reconstruction of the CNN architecture capable of outputting steering angles. The NVIDIA team uses a complex learning procedure with a large amount of data, which is heavily preprocessed. This process is private and is difficult to implement in the context of this scientific work. Therefore, for an accurate analysis, the assumption was made that the processing of the CNN architecture must be fundamentally like that of the real PilotNet model [2]. The model will first be trained on synthetic data generated from the Car Udacity Simulator [17] (Domain 0). The dataset contains images of a video sequence with the respective steering angles executed at that moment of recording. The learned context is formed from features within the images and the output of corresponding steering angles based on these features. Then, the trained models will be evaluated in the digital twin of the car accident scene (Domain 1). Finally, the same model will be used for evaluation in 15 different scenarios where the Structured Domain Randomization process has been examined

(Domain 2).

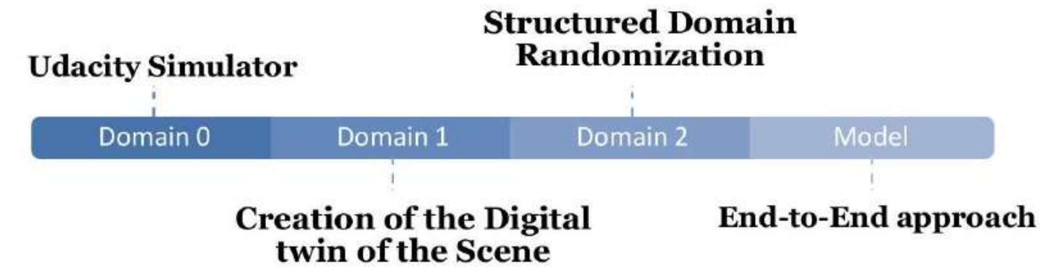

Figure 3.1: Flowchart of methodology.

3.2 Domain 0

The Car Udacity Simulator developed the training and validation domain (domain 0) in this thesis. This simulator provides the capability to emulate vehicle movement in both manual and autonomous modes.

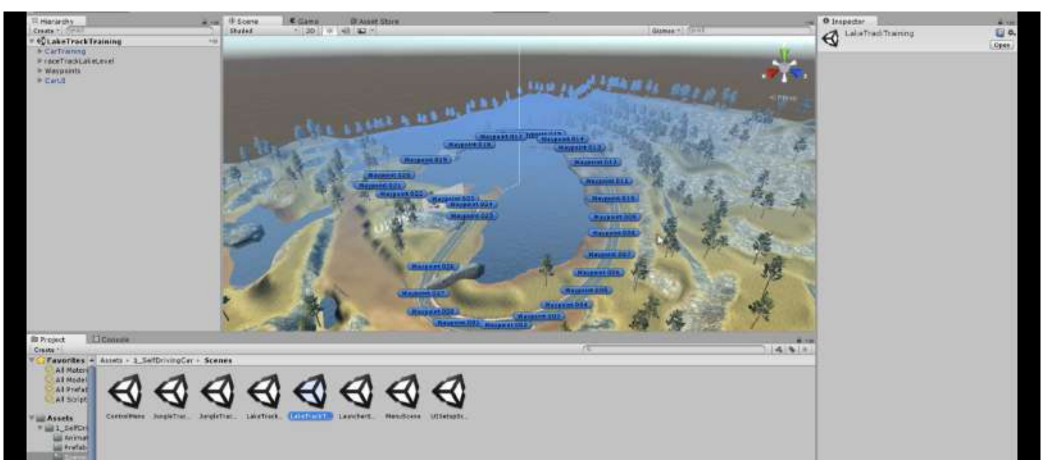

Figure 3.2: Overview of the lake environment inside the Udacity Simulator.

Based on the adaptable nature of the simulator, modifications were made to the pre-existing Lake environment, including the adjustment of rocks as obstacles. These changes were intended to introduce more complex navigational challenges. Approximately 25 minutes of driving were simulated in various styles under manual control. The primary objective of these simulations was to navigate the vehicle without colliding with obstacles or moving down from the track.

Figure 3.3: Overview of the lake environment.

3.2.1 Preparation for the training

The Car Udacity Simulator was employed for the development of the training and validation domain (domain 0) in this thesis. This simulator provides the capability to emulate vehicular movement in both manual and autonomous modes. After that, a series of image augmentation techniques were applied to the generated images to expand the dataset. These augmentation methods included zoom manipulation, panning adjustments, alteration of brightness levels, and flipping operations. These techniques were strategically employed to enhance the diversity of the dataset.

Figure 3.4: Left camera view

Figure 3.5: Center camera view

Figure 3.6: Right camera view

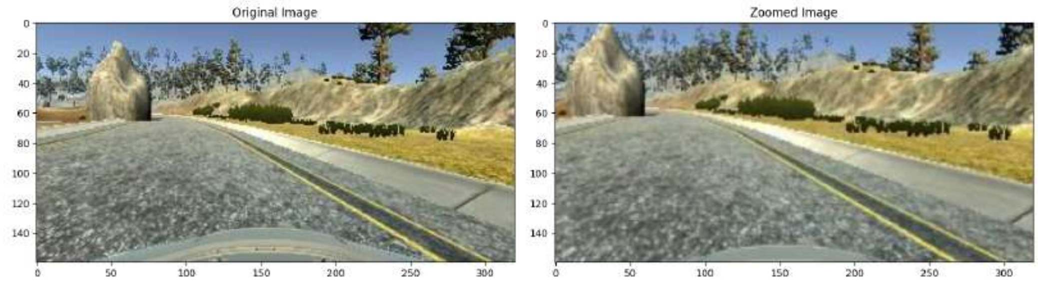

Figure 3.7: Zoom technique

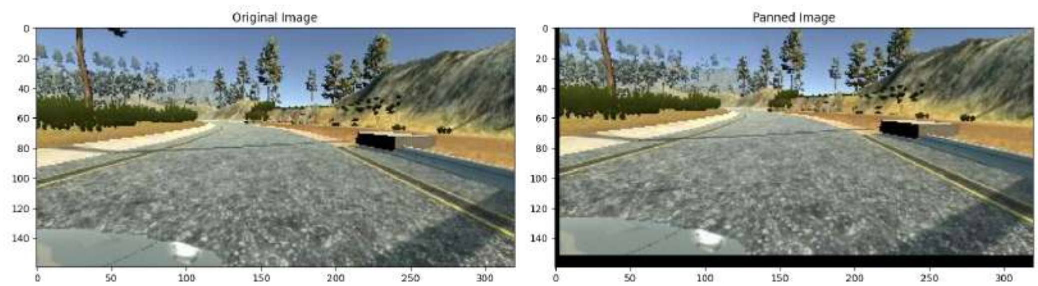

Figure 3.8: Panning technique

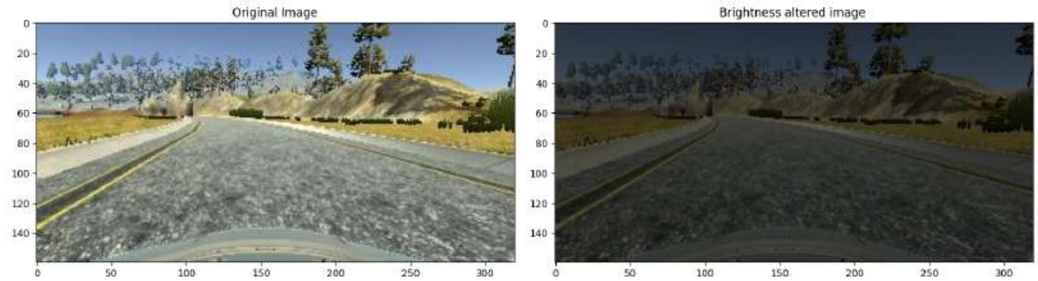

Figure 3.9: Brightness changing technique

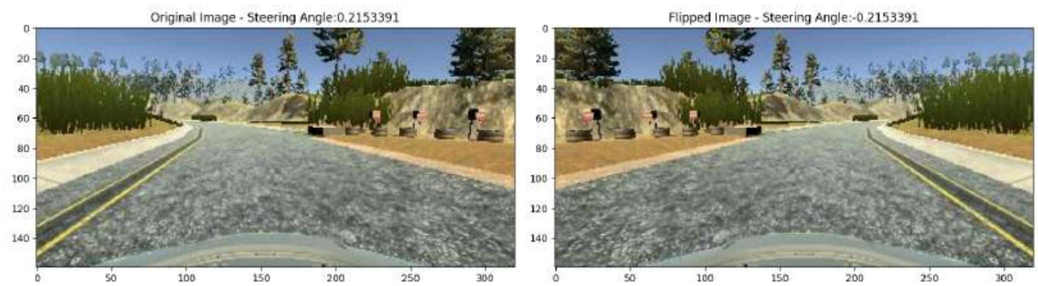

Figure 3.10: Flipping image technique

3.3 Domain 1

Proceeding to Domain 1, this thesis reconstructs the car accident scenario utilizing Unreal Engine 5. This recreation serves as an evaluative testing environment, providing a sophisticated platform for detailed analysis and assessment within the research framework. With its advanced simulation capabilities, the use of Unreal Engine 5 is essential in modeling the accident case, thereby enhancing the validity of the study's findings. In the absence of direct access to the target domain (the real-world environment of the highway), this research necessitated the formulation of certain assumptions. These were based on the available sources of information. This approach was adopted to bridge the gap between the simulated conditions and the actual real-world scenario, ensuring the study's findings' relevance and applicability within the constraints of the available resources. Figure 3.11 and Figure 3.12 showcase the two different angles of the source videos.

Figure 3.11: Source 1
https://www.dailymail.co.uk/sciencetech/article-8377461/Shocking-moment-Telsa-Model-3-Autopilot-mode-crashes-truck-Taiwan-highway.html

Figure 3.12: Source 2
https://www.dailymail.co.uk/sciencetech/article-8377461/Shocking-moment-Telsa-Model-3-Autopilot-mode-crashes-truck-Taiwan-highway.html

3.3.1 Procedural way

The two main assets in this scene that interact directly or indirectly with the ego car are the road and the bridge. Therefore, the creation of these two happened procedurally. The procedural creation of these two assets is pivotal in achieving a high degree of realism and accuracy for the digital twin of the scene.

- For the procedural creation of the highway bridge, the advanced 3D modeling software Blender was used. The process commenced with establishing a foundational geometry representative of the bridge's intended dimensions and spatial orientation within a virtual highway environment. Particular attention was directed towards the design of voids in the bridge's vertical structure. These voids, integral to the architectural integrity of the model, were calibrated to cast realistic shadows onto the road below, mirroring real-world physics. The mirror modifier was employed to ensure symmetrical geometry during the modeling process, achieving the ability to replicate one side of the bridge to the other across the specified axis easily. Also, the Bevel modifier was applied to the bridge's edges to create rounded corners. This is crucial for creating a more lifelike appearance to the bridge, as in the real world, edges are rarely perfectly sharp, contributing to a more detailed rendering of the bridge within the software environment.

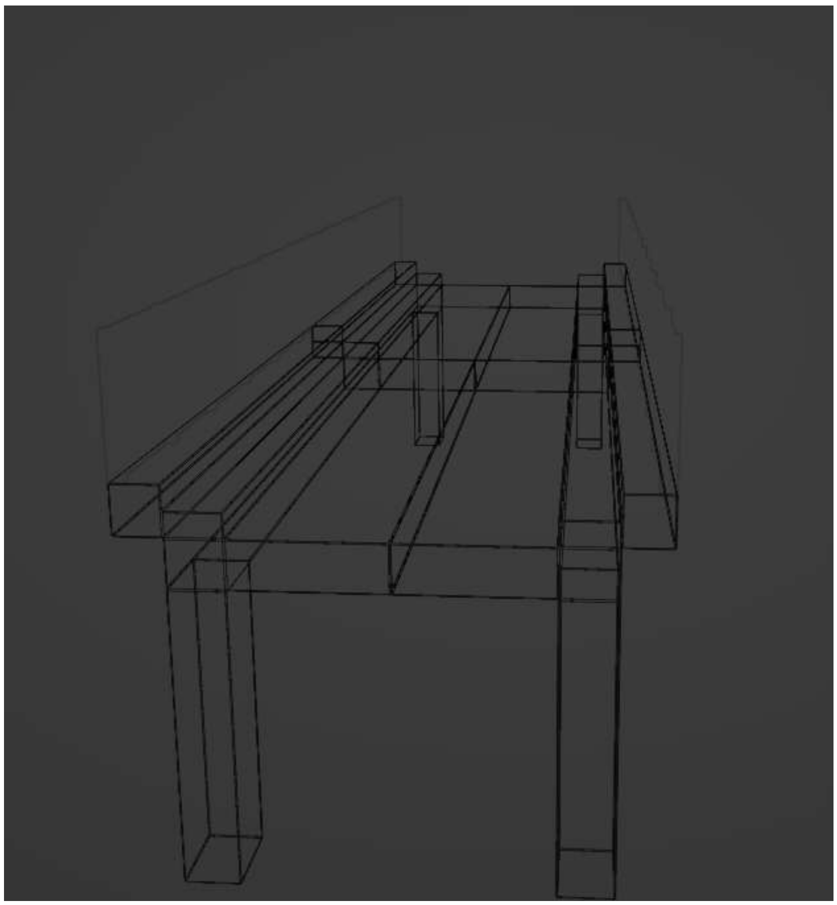

Figure 3.13: Wire edges of the Bridge

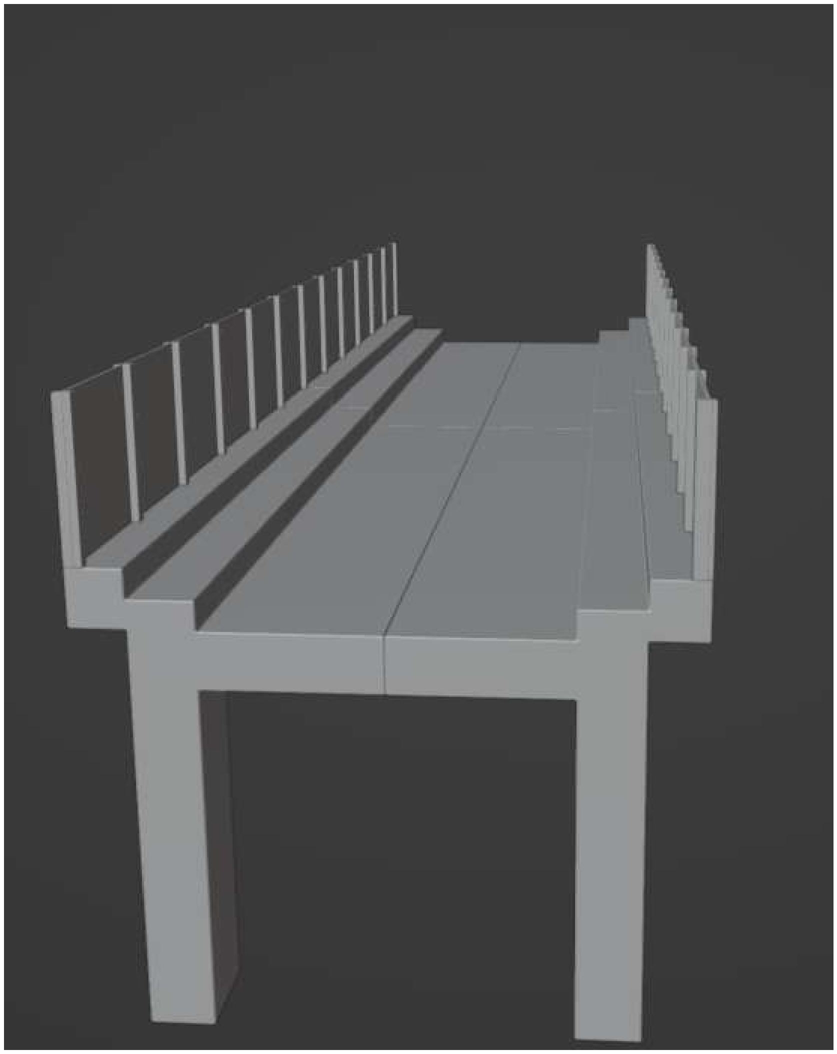

Figure 3.14: Solid depiction of the Bridge

In the next stage of designing the bridge, texturing was applied to impart surface details. They attached the model with material characteristics such as concrete, steel, and glass finishes, thereby completing the visual fidelity of the model in its simulated environment. After the export of the synthetic model from Blender to Unreal Engine 5, the interplay between light and shadow was carefully monitored. Ensuring that the resulting shadow patterns were accurate and dynamic, responding appropriately to simulated environmental lighting conditions.

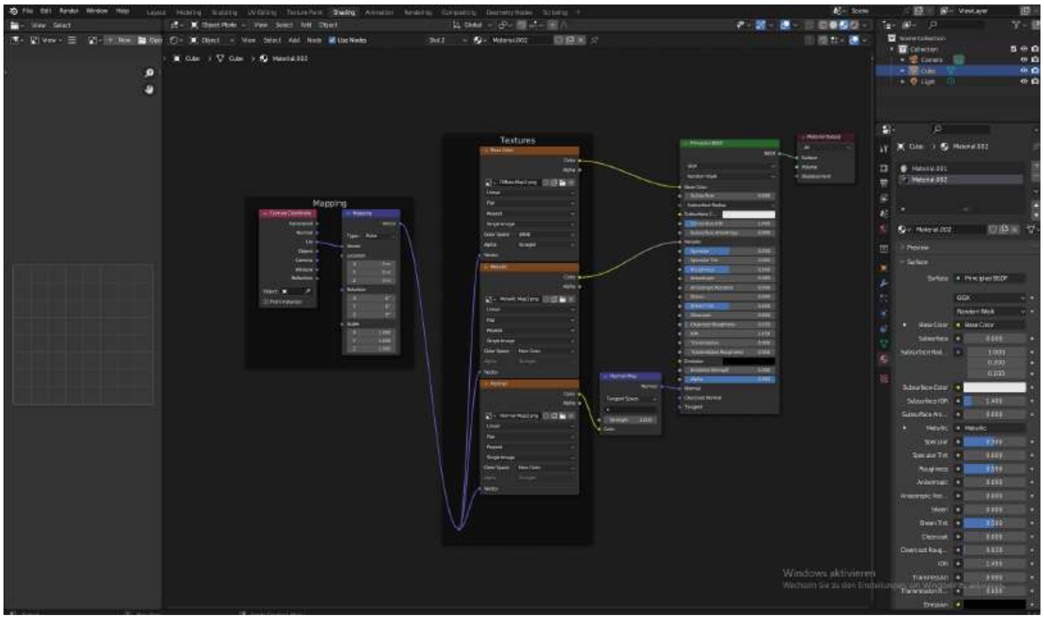

Figure 3.15: Blender implementation of the bridge

Figure 3.16: Final form of the Bridge

- For the procedural creation of the highway road, Blender software was used again. The main source of information about road design was the documentation of Asian Highway Design Standards [5]. Emphasis was placed on several key aspects of road design to enhance realism and functionality. Firstly, the road length was carefully calibrated to reflect typical highway proportions. Secondly, the color and spacing of road markings were precisely replicated, adhering to standard highway specifications.

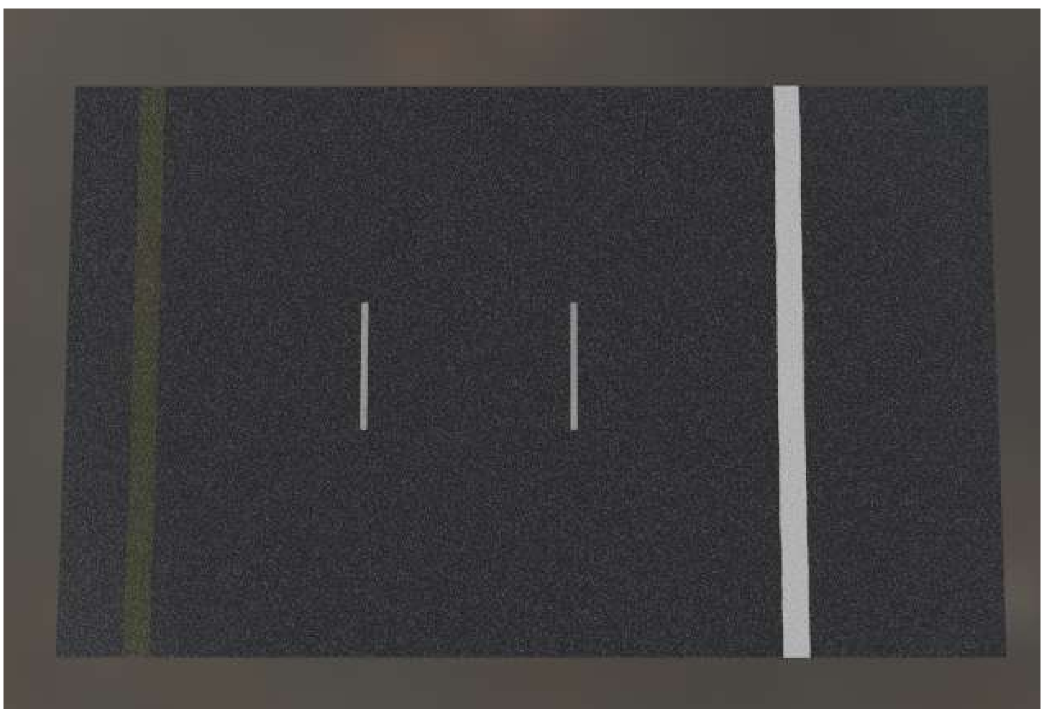

Figure 3.17: Road

Attention was also given to the curvy way of the road, ensuring that it mimicked realistic highway bends and turns. In addition to these elements, safety features such as barriers were added on both sides of the road. All these components were integrated into the Blender model with precision, ensuring that the final render would be visually accurate and representative of real-world highway engineering principles. After the export of the synthetic model from Blender to Unreal Engine 5, the precise placement of the road with the bridge happened.

Figure 3.18: Curvy Way of the Road

3.3.2 Unreal Engine 5 packages

Moving on, all the scene assets were created using different Unreal Engine packages. Ego-car, Truck: The “City Samples” package facilitates a realistic movement and interaction of the vehicle within the virtual space. Biome: The surrounding biome was introduced from the “Megascans Tree” packages, precisely designed using Unreal Engine 5’s advanced rendering capabilities and the new feature of spline creation. Traffic Sign: The placement and design of traffic signs are integral to the authenticity of the highway environment. The package of “Traffic signs” was used.

3.3.3 Combining all the assets in Unreal Engine 5

The final step towards the recreation of the car accident scene inside Unreal Engine 5 was the combination of all the assets. Knowing the unavailability of specific camera details such as width, height, and focal length from the Tesla car involved in the accident, recourse was taken from the comma2k19 dataset [13]. This dataset, widely recognized and utilized in numerous publications in the autonomous driving space, provides real-world data captured from a car's perspective. The fSpy software extracted the necessary camera parameters from the images. These parameters were then replicated in the camera settings within Unreal Engine 5, ensuring an accurate simulation of the accident's visual perspective.

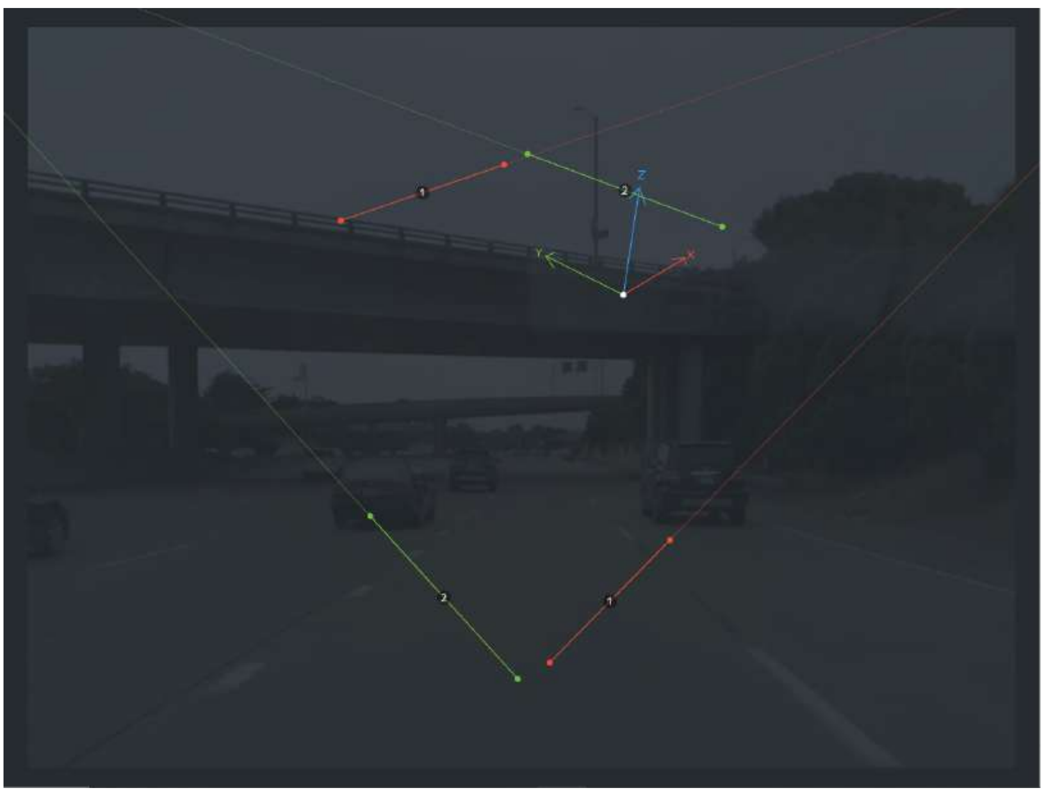

Figure 3.19: FSPY software

Therefore, after adding all the assets together, the attachment of the car with a Camera in a Film sequencer came next.

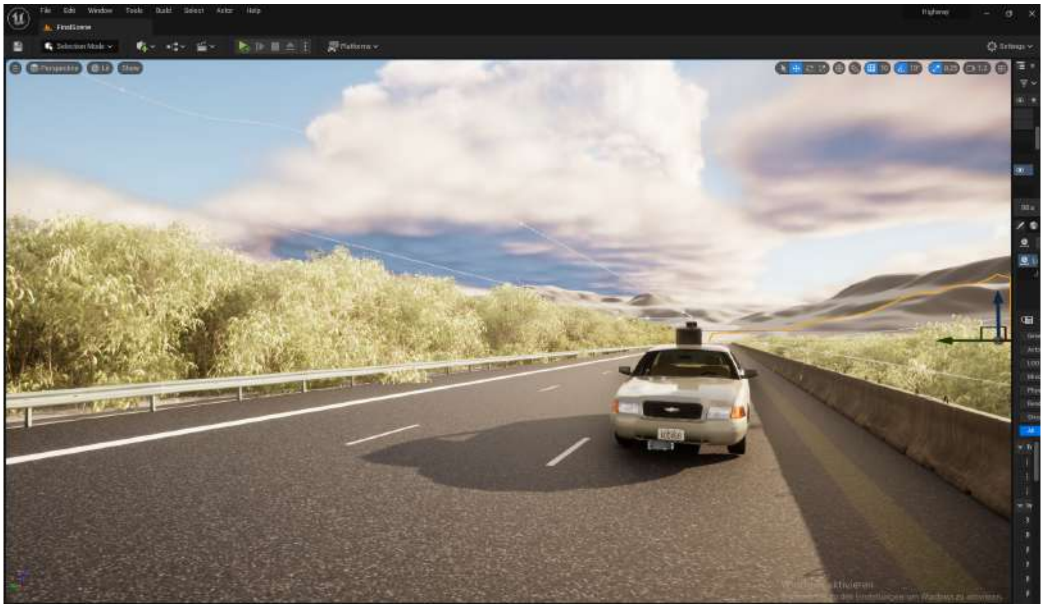

Figure 3.20: Camera attached to the car ready for the filmsequencer

Figure 3.21: Overview of the Original Scenario from the perspective of the car

3.4 Domain 2

Moving on to analyzing the car accident scenario, this thesis takes a step to formulate assumptions grounded in specific observational facts. These include the scarcity of computer vision systems encountering a truck roof on the road, the impact of the shadow cast by the bridge on autopilot reaction, the challenges posed by a curvy road as opposed to a straight path, and the distinct color of the truck, noting the difficulties autopilot systems face in detecting stationary objects. To test these assumptions, alternative scenarios were created, generating various combinations of these elements. These scenarios will be elaborated upon later in the thesis, comprehensively examining their potential impact on the accident.The principle of Structured Domain Randomization underpinned the methodology adopted for creating alternative scenarios. Initially, a set of parameters considered influential to the scene's dynamics and the model's potential response to their modification was identified. Subsequently, fifteen alternative scenarios of the original scene were constructed, predicated on the presence or absence of four specific parameters: (#no_bridge, #alternate_color_of_the_truck(black), #not_curvy_, (straight_road), #not_overturned_truck(upright_truck) . Figure 3.22 presents a comprehensive schematic of the described approach. In the next figures of the subsection, they will be presented in detail with the representation of each scenario.

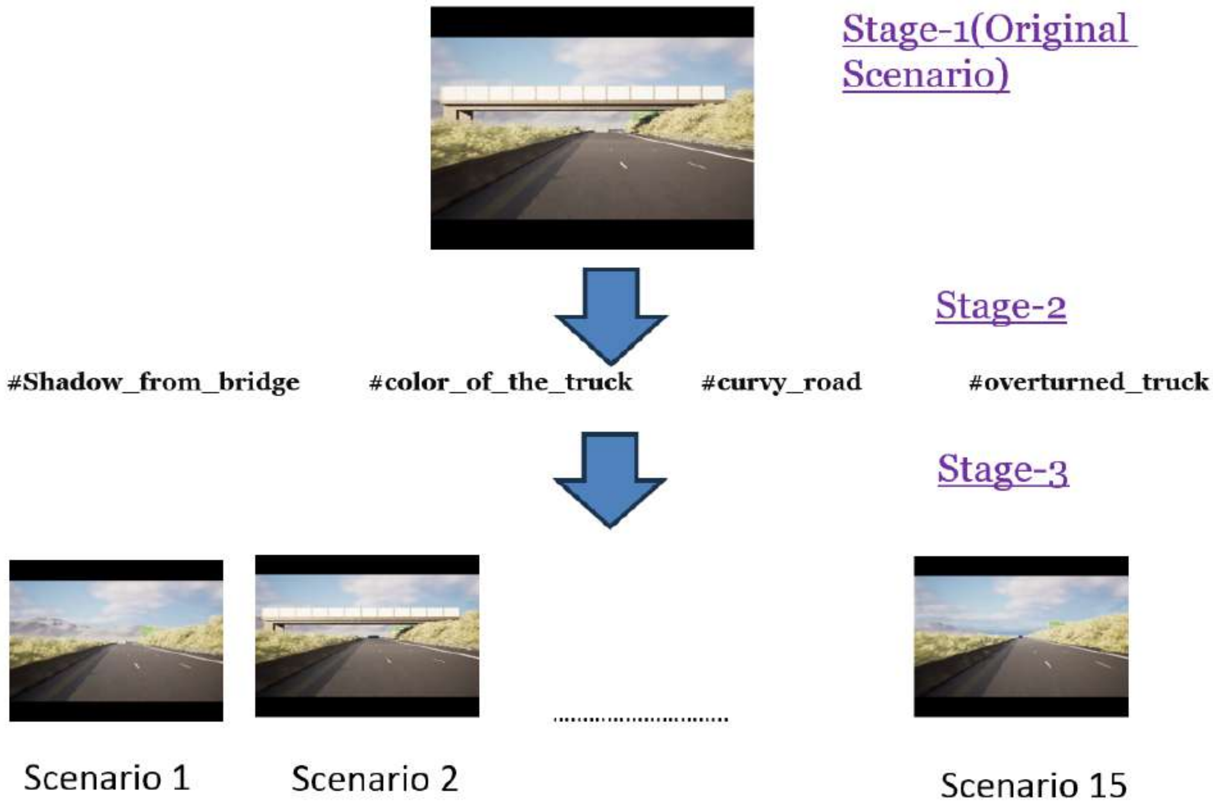

Figure 3.22: Overview of the methodology for the creation of Domain 2

- Scenario1:#no_bridge

Figure 3.23: Frame from Scenario 1

- Scenario2::#alternate_color_of_the_truck(black)

Figure 3.24: Frame from Scenario 2

- Scenario3:#not_curvy_road(straight_road)

Figure 3.25: Frame from Scenario 3

- Scenario4:#not_overturned_truck(upright_truck)

Figure 3.26: Frame from Scenario 4

- Scenario5:#no_bridge and #alternate_color_of_the_truck(black)

Figure 3.27: Frame from Scenario 5

- Scenario6:#no_bridge and #not_curvy_road(straight_road)

Figure 3.28: Frame from Scenario 6

- Scenario7:#no_bridge and #not_overturned_truck(upright_truck)

Figure 3.29: Frame from Scenario 7

- Scenario8:#alternate_color_of_the_truck(black) and #not_curvy_road(straight_road)

Figure 3.30: Frame from Scenario 8

- Scenario9:#alternate_color_of_the_truck(black) and #not_overturned_truck(upright_truck)

Figure 3.31: Frame from Scenario 9

- Scenario10:#not_curvy_road(straight_road) and #not_overturned_truck(upright_truck)

Figure 3.32: Frame from Scenario 10

- Scenario11:#no_bridge and #not_curvy_road(straight_road) and #not_overturned_truck (upright_truck)

Figure 3.33: Frame from Scenario 11

- Scenario12:#no_bridge and #alternate_color_of_the_truck(black) and #not_curvy_road (straight_road)

Figure 3.34: Frame from Scenario 12

- Scenario13:#no_bridge and #alternate_color_of_the_truck(black) and #not_overturned_truck (upright_truck)

Figure 3.35: Frame from Scenario 13

- Scenario14:#alternate_color_of_the_truck(black) and #not_curvy_road(straight_road) and #not_overturned_truck(upright_truck)

Figure 3.36: Frame from Scenario 14

- Scenario15:#no_bridge and #alternate_color_of_the_truck(black) and #not_curvy_road (straight_road) and #not_overturned_truck(upright_truck)

Figure 3.37: Frame from Scenario 15

3.5 Model

The model used in this thesis is a convolutional neural network (CNN) inspired by NVIDIA's architecture. The input to the network is an image of size 66x200 pixels with three channels (RGB). Before being fed into the network, images undergo preprocessing, including normalization and other transformations to enhance features relevant to driving. In the feature extraction through the convolutional layers part of the network, the first layer applies 24 filters of size 5x5 to the input image. Then, each filter convolves over the image to extract specific features. The stride of 2 reduces the spatial dimensions of the output, focusing on more prominent features and reducing computational load. Regarding the depth of Filters, subsequent convolutional layers with 36, 48, and 64 filters increase the depth of the network, allowing it to capture more complex and abstract features. These layers can recognize patterns such as lane markings, other vehicles, and road signs. Moving on, the Exponential Linear Unit (ELU) activation function introduces nonlinearity, enabling the network to model complex relationships in the data.

ELU helps in learning faster and reduces the problem of vanishing gradients. After extracting spatial hierarchies of features, the data is flattened into a one-dimensional vector. This transformation prepares the data for the fully connected dense layers. With decreasing numbers of neurons (100, 50, and 10), these layers integrate the extracted features to form a high-level understanding of the current driving scenario. The final layer(output layer), a single neuron, indicates that the network outputs a continuous variable. The steering angle directly correlates the image data with a specific driving action.

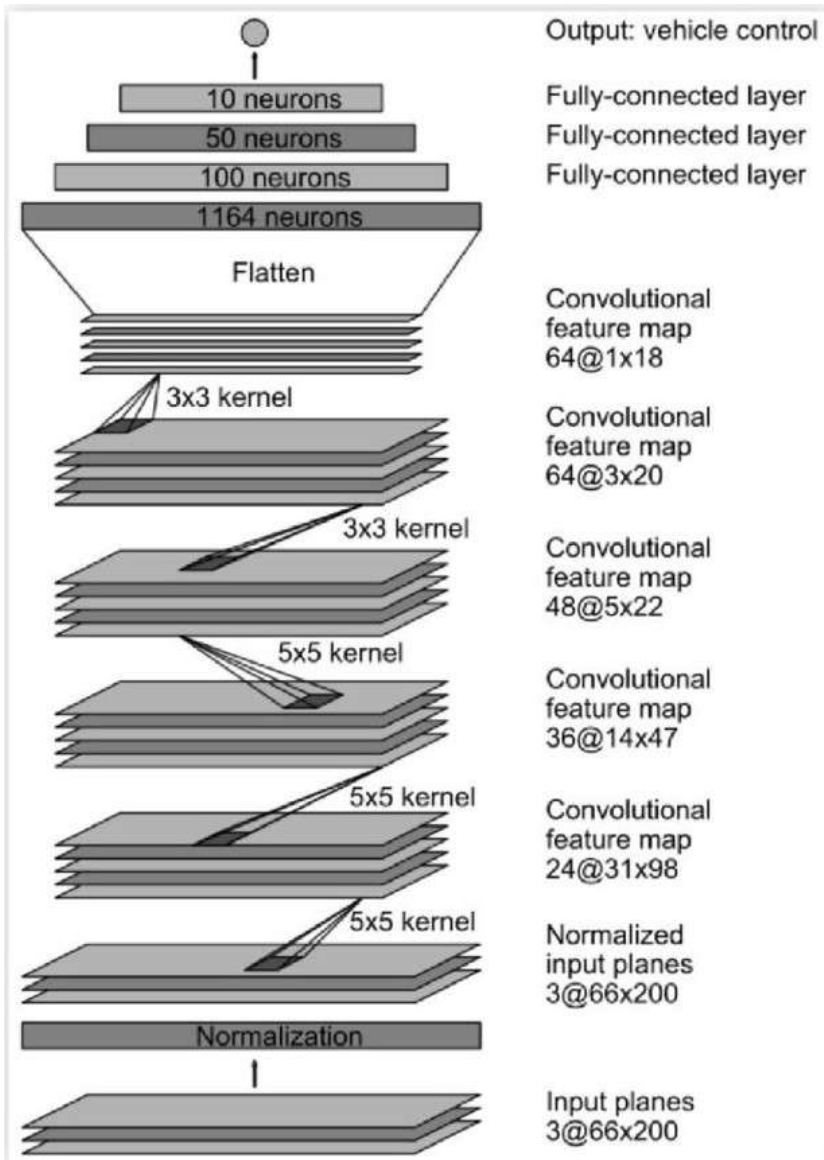

Figure 3.38: NVIDIAs-PilotNet-architecture
https://syncedreview.com/2017/05/28/how-a-deep-neural-network-trained-with-end-to-end-learning-steers-a-car/

3.5.1 Training Process

The model is trained with a custom batch generator function, which dynamically feeds batches of training data (X_train, y_train) into the model. The training set comprises 300 steps per epoch, indicating the number of batches processed in each training epoch. The model is trained for ten epochs, allowing the neural network to iteratively adjust and improve its parameters based on the provided data. Additionally, validation is performed using a separate batch generator for validation data (X_validation, y_validation), with 200 validation steps per epoch. This validation process is crucial for assessing the model's performance on data it has not seen during training. A parameter is set to 1, enabling the output of training progress for each epoch and providing insights into the learning process. Another parameter ensures that the training data is shuffled before each epoch, promoting the model's ability to generalize and preventing overfitting to the order of the data. This training procedure is essential to optimize the model's intended application within the thesis.

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 31, 98, 24)	1824
conv2d_5 (Conv2D)	(None, 14, 47, 36)	21636
conv2d_6 (Conv2D)	(None, 5, 22, 48)	43248
conv2d_7 (Conv2D)	(None, 1, 18, 64)	76864
flatten_1 (Flatten)	(None, 1152)	0
dense_4 (Dense)	(None, 100)	115300
dense_5 (Dense)	(None, 50)	5050
dense_6 (Dense)	(None, 10)	510
dense_7 (Dense)	(None, 1)	11

Total params: 264443 (1.01 MB)
Trainable params: 264443 (1.01 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 3.39: Training process 1

```
  history = model.fit_generator(batch_generator(X_train, y_train, 100, 1),
Epoch 1/10
300/300 [==============================] - 209s 692ms/step - loss: 0.3673 - val_loss: 0.2752
Epoch 2/10
300/300 [==============================] - 194s 648ms/step - loss: 0.2793 - val_loss: 0.2632
Epoch 3/10
300/300 [==============================] - 187s 624ms/step - loss: 0.2649 - val_loss: 0.2361
Epoch 4/10
300/300 [==============================] - 187s 623ms/step - loss: 0.2541 - val_loss: 0.2319
Epoch 5/10
300/300 [==============================] - 186s 620ms/step - loss: 0.2399 - val_loss: 0.2009
Epoch 6/10
300/300 [==============================] - 188s 626ms/step - loss: 0.2286 - val_loss: 0.2003
Epoch 7/10
300/300 [==============================] - 188s 628ms/step - loss: 0.2210 - val_loss: 0.1803
Epoch 8/10
300/300 [==============================] - 187s 625ms/step - loss: 0.2138 - val_loss: 0.1894
Epoch 9/10
300/300 [==============================] - 188s 626ms/step - loss: 0.2078 - val_loss: 0.1771
Epoch 10/10
300/300 [==============================] - 190s 635ms/step - loss: 0.2028 - val_loss: 0.1692
```

Figure 3.40: Training process 2

Chapter 4

Results

The following chapter will analyze and explain the computational and qualitative results after explaining the general rules. From the observed correlations, conclusions will be drawn about the driving behavior of our model in every scenario. Furthermore, based on the comparison of the results, logical assumptions will be made regarding the causes of a possible accident in this highway environment.

4.1 General rules

To start the results analysis, the general rules should first be highlighted. Figure 4.1 illustrates a road segment divided into six distinct zones. Each zone is characterized by a specific range of steering angle values, providing a framework for evaluating the vehicle's steering behavior. This zoning approach facilitates a structured steering response analysis across different road segments. More precisely:

- Zone 1: Steering angle value ≥ -0.166 and Steering angle value < 0
- Zone 2: Steering angle value ≥ 0 and Steering angle value < 0.166
- Zone 3: Steering angle value ≥ 0.166 and Steering angle value < 0.322
- Zone 4: Steering angle value ≥ 0.322 and Steering angle value < 0.498
- Zone 5: Steering angle value ≥ 0.498 and Steering angle value < 0.664
- Zone 6: Steering angle value ≥ 0.664 and Steering angle value < 0.833

When the predicted steering angles fall below -0.166 or exceed 0.833, they are classified as exhibiting unacceptable behavior. This classification is based on the predefined operational parameters for the steering mechanism within the Udacity environment. This intolerable behavior causes an off-road incident or a crash with the barriers. In all scenarios, the ego-car's starting point is in the middle of Zone 1 and Zone 2. In the Original Scenario and all the other scenarios where the #up_right_truck parameter is not included, for the ego car to avoid the accident, the predicted steering angle in the last frame should be equal and higher than 0.322(Inside Zone 4, Zone 5, Zone 6).On the other hand, in scenario four and all the other scenarios where they include the #up_right_truck parameter, the predicted steering angle in the last frame should be equal and higher than 0.166 (Inside Zone 3, Zone 4, Zone 5, Zone 6).

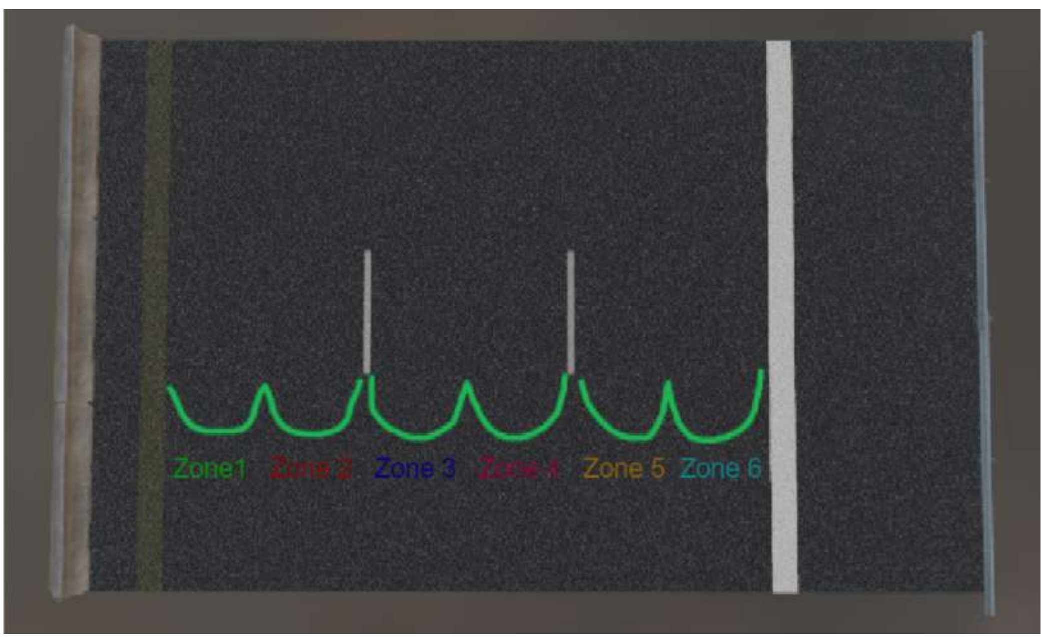

Figure 4.1: The Zones

4.2 Computational results

The evaluation metrics were established for the analysis after getting the predicted steering angles for all 16 different scenarios. The mean of steering angles on every scenario provides a fundamental measure of the central tendency in steering behavior, offering valuable insights into the system's overall performance and tendencies. The mean of steering angles metric is essential for evaluating autonomous driving systems' accuracy, safety, and reliability. However, it should be interpreted in conjunction with other metrics. In this regard, the smoothness metric gives insights into the nature of steering adjustments over time, necessary for steering control, while the standard deviation metric provides an understanding of the variability in steering behavior, crucial for evaluating the vehicle's adaptability and reliability in different driving scenarios.

- Mean of Steering Angles = $\frac{1}{N}\sum_{i=1}^{N}\theta_i$
- Standard Deviation = $\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\theta_i - \mu)^2}$
- Smoothness = $\frac{1}{\text{mean}(|\Delta\theta|)}$

Drawing from the results presented in Figure 4.2, the scenarios can be categorized according to their mean value similarities. This grouping facilitates a more coherent analysis of the metric, offering valuable perspectives on the performance of the autonomous driving model.

Scenario	Mean of steering angles
Original	0.433
Scenario 1	0.405
Scenario 2	0.312
Scenario 3	0.674
Scenario 4	0.421
Scenario 5	0.286
Scenario 6	0.646
Scenario 7	0.408
Scenario 8	0.56
Scenario 9	0.346
Scenario 10	0.609
Scenario 11	0.601
Scenario 12	0.358
Scenario 13	0.342
Scenario 14	0.563
Scenario 15	0.549

Figure 4.2: Mean of average

Group 1: Moderate Right Bias
Original Scenario (0.4336), Scenario 1 (0.4059), Scenario 4 (0.4219), Scenario 7 (0.4080) These scenarios have mean steering angles in close range, indicating a moderate rightward steering bias. This might suggest driving conditions with a fair mix of straight paths and some right-biased turns. The driving model appears to perform consistently in these conditions. Regarding the driving model performance, it indicates the model's ability to handle typical driving scenarios with a blend of straight and curved paths.

Group 2: Slight Right Bias

Scenario 2 (0.3122), Scenario 5 (0.2865), Scenario 9 (0.3462), Scenario 12 (0.3584), Scenario 13 (0.3427) This group of scenarios has lower mean values, suggesting a slight rightward steering bias. This group could represent scenarios with a balance of left and right turns but with little inclination to turn toward the right. The driving model performance in this group shows a balanced approach to steering, handling both left and right turns with a slight preference for right.

Group 3: Strong Right Bias

Scenario 3 (0.6741), Scenario 6 (0.6468), Scenario 8 (0.5604), Scenario 10 (0.6092), Scenario 11 (0.6018), Scenario 14 (0.5632), Scenario 15 (0.5491) This group shows a robust rightward steering bias with significantly higher mean values. Therefore, the driving model's performance in these scenarios suggests a strong tendency to make right turns.

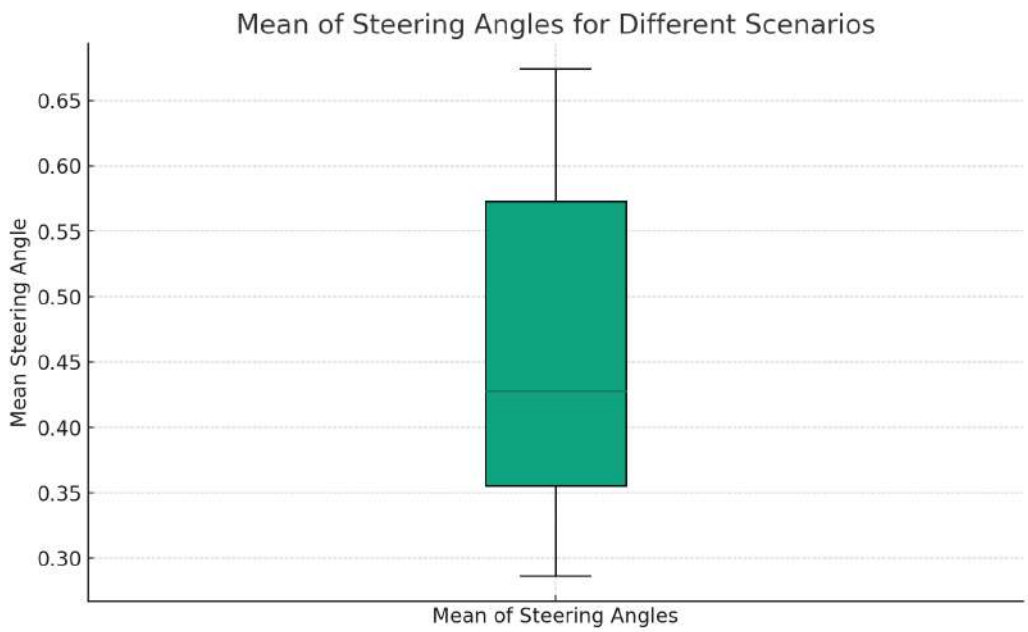

Figure 4.3: Box-plot of Mean of steering angles

After the grouping and analysis of the mean average of steering angles, it is clear the model's ability to adapt to varying driving conditions also reveals a general tendency towards right-biased steering. Another evaluation metric, standard deviation, measures the amount of variability or spread in the steering angles from their mean value. A higher standard deviation indicates more significant variability, while a lower standard deviation suggests more consistency in the steering behavior.

Grouping Based on Similar Standard Deviation:

Scenario	Standard deviation of steering angles
Original	0.266
Scenario 1	0.127
Scenario 2	0.251
Scenario 3	0.202
Scenario 4	0.219
Scenario 5	0.122
Scenario 6	0.052
Scenario 7	0.06
Scenario 8	0.259
Scenario 9	0.242
Scenario 10	0.234
Scenario 11	0.068
Scenario 12	0.102
Scenario 13	0.075
Scenario 14	0.266
Scenario 15	0.151

Figure 4.4: Standard Deviation Steering Angles table

Group 1: High Variability

Original Scenario (0.2665), Scenario 2 (0.2515), Scenario 8 (0.2592), Scenario 9 (0.2427), Scenario 10 (0.2342), Scenario 14 (0.2661). These scenarios exhibit a higher standard deviation, indicating a wide range of steering angles. High variability suggests the model can have a wide range of steering responses.

Group 2: Moderate Variability

Scenario 3 (0.2022), Scenario 4 (0.2192), Scenario 15 (0.1517). This group shows a moderate level of variability. Indicates good adaptability to varied driving conditions.

Group 3: Low Variability

Scenario 1 (0.1278), Scenario 5 (0.1223), Scenario 6 (0.0527), Scenario 7 (0.0601), Scenario 11 (0.0681), Scenario 12 (0.1026), Scenario 13 (0.0753). Scenarios in this group have the lowest standard deviation, suggesting consistent steering behavior. This means that the model in this group is operating in more predictable and stable conditions, with less need for frequent or significant steering adjustments.

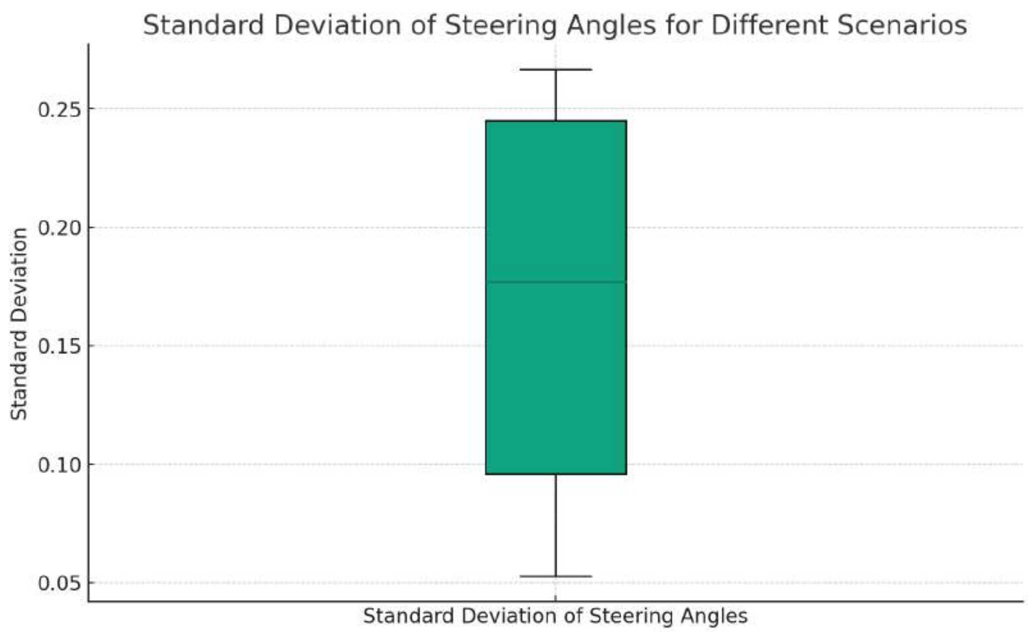

Figure 4.5: Box-plot of Standard deviation

Another metric, smoothness, is defined here as the inverse of the average rate of change of steering angles. Higher smoothness values indicate more gradual changes in steering angles, suggesting smoother driving behavior. Conversely, lower smoothness values suggest more frequent or abrupt changes in steering.

Grouping Based on Similar Smoothness values :

Scenario	Smoothness
Original	21.03
Scenario 1	63.87
Scenario 2	20.05
Scenario 3	22.88
Scenario 4	21.13
Scenario 5	62.43
Scenario 6	67.11
Scenario 7	69
Scenario 8	22.32
Scenario 9	21.3
Scenario 10	22.13
Scenario 11	68.25
Scenario 12	74.05
Scenario 13	71.8
Scenario 14	20.81
Scenario 15	56.51

Figure 4.6: Smoothness of Steering angles

Group 1: High Smoothness

- Scenario 1 (63.8740)

- Scenario 5 (62.4320)
- Scenario 6 (67.1193)
- Scenario 7 (68.9936)
- Scenario 11 (68.2582)
- Scenario 12 (74.0506)
- Scenario 13 (71.8037)
- Scenario 15 (56.5178)

These scenarios exhibit very high smoothness, indicating that the steering changes are gradual and smooth. Indicates the model's capability to make smooth, gradual steering adjustments. This is indicative of a stable and comfortable driving experience.

Group 2: Moderate Smoothness

- Original Scenario (21.0375)
- Scenario 2 (20.0586)
- Scenario 3 (22.8890)
- Scenario 4 (21.1353)
- Scenario 8 (22.3229)
- Scenario 9 (21.3061)
- Scenario 10 (22.1323)
- Scenario 14 (20.8115)

This group shows moderate levels of smoothness, indicating a balance between straight and curved driving paths. Reflects a more varied steering behavior, combining rapid and gradual steering adjustments.

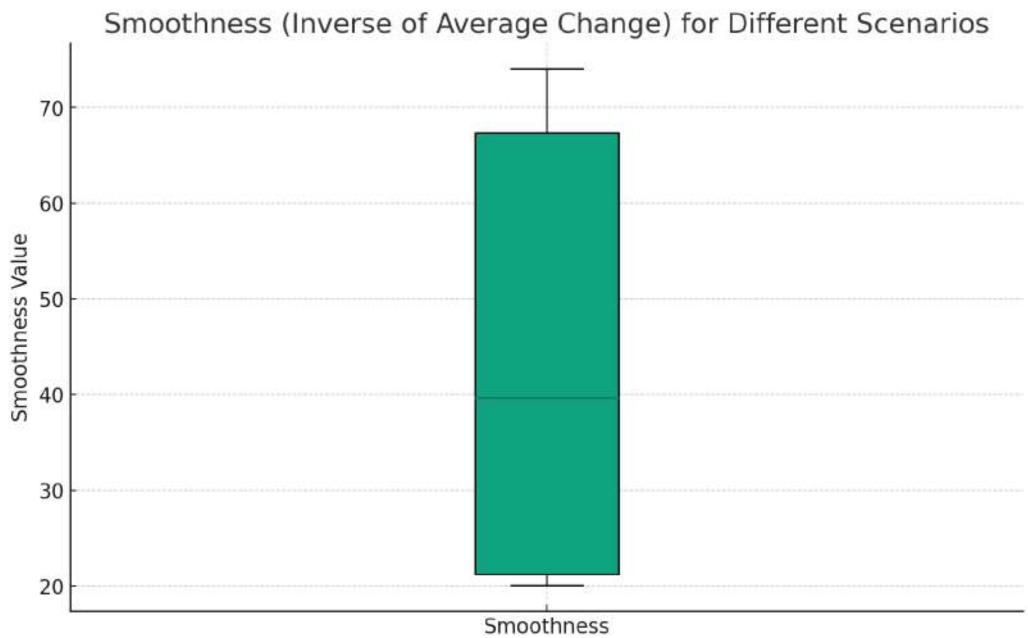

Figure 4.7: Box-plot of smoothness

Finally, another critical evaluation metric is accuracy, defined by the type:

$$\text{accuracy} = \left(\frac{\text{number of images with acceptable trajectory}}{\text{total number of images}}\right) \times 100$$

Scenario	#collision	#off_roads	accuracy
Original	0	4	97%
Scenario 1	0	1	99%
Scenario 2	1	3	97%
Scenario 3	8	6	90%
Scenario 4	0	3	98%
Scenario 5	1	1	98.5%
Scenario 6	2	0	98.5%
Scenario 7	0	0	100%
Scenario 8	1	13	90%
Scenario 9	3	1	97%
Scenario 10	1	0	99%
Scenario 11	0	0	100%
Scenario 12	1	0	99%
Scenario 13	1	0	99%
Scenario 14	1	13	90%
Scenario 15	0	0	100%

Figure 4.8: Smoothness of Steering angles

In all the scenarios, the model achieves accuracy higher or equal to 90%. The highest accuracy(100%) has been achieved in Scenario 7,11,15, wherein all of them the parameters #no_bridge and #up_right_truck were included. On the other hand, the accuracy table reveals that scenarios 3, 8, and 14 manifest the lowest accuracy percentages for the model's responses. Upon examining the results of Scenario 3 in relation to Scenarios 8 and 14, which all share the #straight_road parameter, it can be concluded that this parameter creates a setting in which the model is prone to generating a significant number of unacceptable predicted steering angles. Although these predictions are directed correctly (to the right), their values exceed the established acceptable threshold.

4.3 Qualitative results

Scenario	safety
Original	1
Scenario 1	1
Scenario 2	0
Scenario 3	1
Scenario 4	1
Scenario 5	0
Scenario 6	1
Scenario 7	1
Scenario 8	1
Scenario 9	0
Scenario 10	0
Scenario 11	1
Scenario 12	0
Scenario 13	0
Scenario 14	1
Scenario 15	1

Figure 4.9: Safety table

Grouping based on Safety Metric: Group 1: The car avoids the crash with the truck Original Scenario, Scenario 1, Scenario 3, Scenario 4, Scenario 6, Scenario 7, Scenario 8, Scenario 11, Scenario 14, Scenario 15

Group 2: The car does not avoid the crash with the truck Scenario 2, Scenario 9, Scenario 5, Scenario 10, Scenario 12, Scenario 13.

An analysis of the Safety table indicates that scenarios 2, 5, 9, 10, 12, and 13 result in a collision with the truck. A further investigation into these scenarios reveals that five of the six incorporate the #black_truck parameter. This observation leads to the assumption that the black color of the truck may be confusing within the system, leading to the prediction of unacceptable steering angles. Additionally, three of these six scenarios involve the combined parameters of the #no_bridge and #black_truck parameters, suggesting a potential interaction effect between these factors.

Upon evaluating the Accuracy and Safety table, it becomes evident that scenarios 7, 11, and 15 are where the model demonstrates perfect accuracy and maintains complete safety, as indicated by the metrics. These scenarios all share the inclusion of #no_bridge and #up_right_truck parameters, leading to the inference that their combination creates a safer operational context for the model. However, Scenario 13, despite sharing those two parameters, shows a slightly reduced accuracy of 99% Finally, upon examination of both the Accuracy and Safety tables, it is observed that Scenario 9 exhibits the lowest accuracy rate amongst the six scenarios categorized as unsafe.

4.4 Comparing scenarios

4.4.1 Original Scenario-Scenario 1

Because only one parameter changed, it is safe to say that this parameter change is responsible for the differences. In both scenarios, the model avoids a crash with the truck. But one difference is that when the car enters the area down the bridge in the original scenario, two off-road incidents are created, unlike in scenario 1, where the bridge parameter does not exist. These two off-road incidents in these frames don't exist.

Figure 4.10: Original Scenario frame

Figure 4.11: Scenario 1 frame

Figure 4.12: Original Scenario frame

Figure 4.13: Scenario 1 frame

4.4.2 Original Scenario-Scenario 2

Again, because only one parameter changed, it is safe to assume that this parameter change is responsible for the differences we have. In the original scenario, the accident is avoided, unlike the case of the actual Scenario. This leads us to assume that the change of the color to black causes this crash in the last frame. Looking more precisely at our predicted steering angles, it is clear that in the original scenario, our model predicts taking a far-right turn to avoid the truck, unlike in Scenario 2, where the model predicts going left.

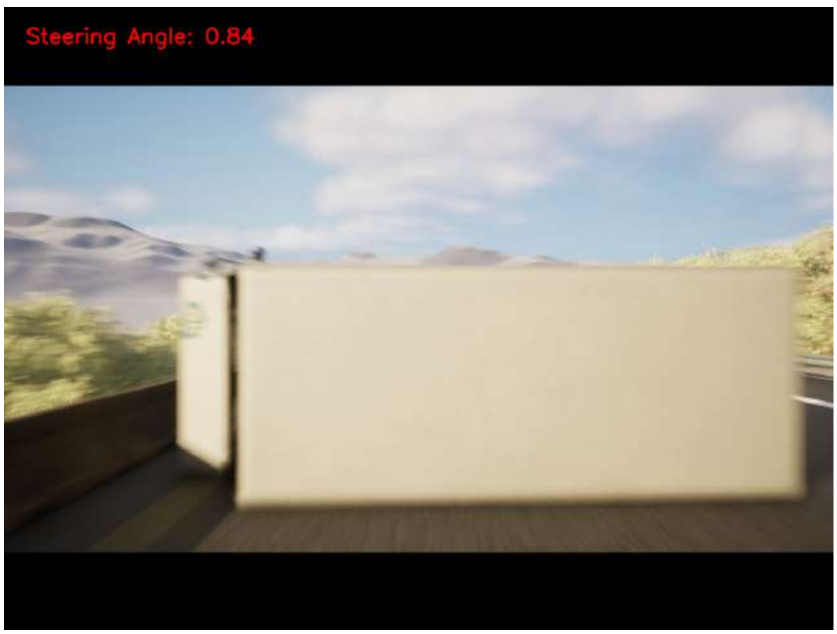

Figure 4.14: Original Scenario frame

Figure 4.15: Scenario 2 frame

4.4.3 Original Scenario-Scenario 3

Similar to the previous comparisons, because only one parameter changed, it is safe to claim that this parameter change is responsible for the different results. In both scenarios, the model avoids a crash with the truck. But in scenario three, many far-right predicted steering angles were produced, causing eight collisions with the barriers and six off-road incidents. For example, the following figures show frames that the model predicted steering angle 1, leading to collisions with the obstacles.

Figure 4.16: Frame from Scenario 3

Figure 4.17: Frame from Scenario 3

Figure 4.18: Frame from Scenario 3

Figure 4.19: Frame from Scenario 3

Figure 4.20: Frame from Scenario 3

Figure 4.21: Frame from Scenario 3

Figure 4.22: Frame from Scenario 3

Figure 4.23: Frame from Scenario 3

4.4.4 Original Scenario-Scenario 4

Finally, in this comparison, because only one parameter changed, it is safe to say that this parameter change is responsible for the differences we observe in those two scenarios. Despite the model avoiding the truck in both scenarios, the last predicted steering angle in the Original Scenario causes an off-road incident, unlike the case of Scenario 4, where the predicted steering angle is much smoother.

Figure 4.24: Scenario 4 frame

Figure 4.25: Original Scenario frame

Chapter 5

Summary

This thesis successfully generated high-precision, photorealistic synthetic data using Unreal Engine 5 (UE5) despite lacking direct access to target domain data. In this way, it created an evaluation environment to examine the End-to-End process in autonomous driving through the lens of imitation learning in a highway accident. Based on these simulations, the thesis provided insightful assumptions about the highway accident, contributing to the understanding and patterning of autonomous vehicle behavior in various driving scenarios. In more detail, the influence of the environment parameters like #no_bridge and #alternate_color_of_the_truck(black) and #not_curvy_road (straight_road) and #not_overturned_truck(upright_truck) in the reaction of the driving model, has been investigated. The following are the most representative findings that are derived from this thesis:

- A detailed examination uncovers that 5 out of 6 scenarios where the model does not prevent the truck crash involve the #black_color parameter. This correlation leads to the hypothesis that the black color of the truck creates frustration within the system, resulting in the prediction of unacceptable steering angles.
- The scenarios that stand out in both the Accuracy and Safety table with full accuracy and complete safety share the common parameters of #no_bridge and #up_right_truck. The assumption drawn is that the combination of these two parameters creates a safer operational context for the model.
- All the scenarios that exhibit the lowest accuracy percentages include

the #straight_road parameter.

- The autonomous driving model that is used during the 15 scenarios experiments achieved 60% on safety measurement, and in all of the fifteen scenarios achieved 90% accuracy as defined in the thesis.These percentages indicate that the experimental imitation learning approach that has been followed is reliable for scientific work.

For future updates, it is clear that the variety of parameters that can be changed with the help of Unreal Engine 5 is unlimited. Therefore, the spectrum of alternative scenarios that can be created is an exciting research domain. As far as the model is concerned, a more sophisticated model can separate some of the responsibilities of the autonomous driving process. In the concluding phase of this thesis, an experimental add-on was introduced to the existing model by integrating computer vision techniques for lane detection. After a series of processes, including converting images to grayscale, applying Gaussian blur for image smoothing, utilizing Canny edge detection for identifying edges, implementing region of interest masking, and employing the Hough transform for line detection. An experimental outcome of this upgrade was the successful elimination of off-road behaviors, and the reduction of high-value predicted steering angles in scenario 3, which previously exhibited the highest incidence of such issues.

Figure 5.1: Example from Scenario 3

Figure 5.2: Example from Scenario 3

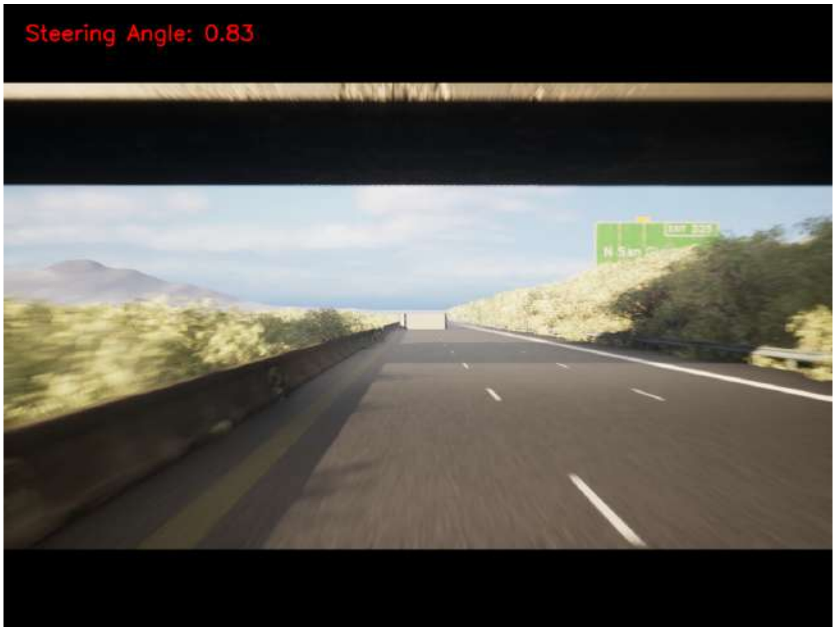

Figure 5.3: Example from Scenario 3

Figure 5.4: Example from Scenario 3

Bibliography

[1] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 2016.

[2] Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. Explaining how a deep neural network trained with end-to-end learning steers a car, 2017.

[3] Lixin Duan, Dong Xu, and Ivor Tsang. Learning with augmented features for heterogeneous domain adaptation, 2012.

[4] IPG Automotive. Vehicle-in-the-loop — ipg automotive. `https://www.ipg-automotive.com/`, 2023.

[5] f Mr. Ishtiaque Ahmed Julian T.H. Kwong. *Asian Highway Design Standard for Road Safety: Design Guidelines*. Korean Expressway Corporation (KEC), 2017.

[6] Helena Kovandová, Jiří Krejčí, and Jiří Kovanda. Crash tests of vehicles – selected experiments and analysis. *VOLUME 6, NUMBER 4*, 2013.

[7] Stacy Liberatore and Lauren Edmonds. Shocking moment tesla model 3 'on autopilot mode' crashes into a truck on taiwan highway. Daily Mail, 2020.

[8] N. Omidi, M. Jafari Eskandari, and M. R. Omidi. Ranking of factors influencing injury and death by accident using analytical approach: A case study of ilam province roads. *Health in Emergencies and Disasters Quarterly*, 5(4):207–214, 2020.

[9] Darsh Parekh, Nishi Poddar, Aakash Rajpurkar, Manisha Chahal, Neeraj Kumar, Gyanendra Prasad Joshi, and Woong Cho. A review on autonomous vehicles: Progress, methods and challenges. *Electronics*, 11(14):2162, 2022.

[10] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017.

[11] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data, 2020.

[12] Francisco Rosique, Pedro J. Navarro, Cristina Fernández, and Antonio Padilla. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors (Basel, Switzerland)*, 19(3), 2019.

[13] Harald Schafer, Eder Santana, Andrew Haden, and Riccardo Biasini. A commute in data: The comma2k19 dataset, 2018.

[14] Tesla. *Model 3 Owner's Manual: About Autopilot - How It Works.* Tesla, 2022.

[15] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world, 2017.

[16] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization, 2018.

[17] Udacity. An open source self-driving car, 2017.

[18] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data, 2022.